AWS ホワイトペーパー

AWS でのワークロードの災害対策: クラウド内での復旧

AWS でのワークロードの災害対策: クラウド内での復旧: AWS ホワイトペーパー

Table of Contents

AWS でのワークロードの災害対策: クラウド内での復旧

発行日: 2021 年 2 月 12 日 (ドキュメントの改訂)

要約

災害対策は、災害に備え、災害から復旧するプロセスです。ワークロードやシステムがプライマリのデプロイ先でビジネス目標を達成できなくなるようなイベントは、災害と見なされます。このホワイトペーパーでは、AWS にデプロイしたワークロードの災害対策を計画およびテストするためのベストプラクティスを紹介し、リスクを軽減して、ワークロードの目標復旧時間 (RTO) と目標復旧時点 (RPO) を満たすさまざまなアプローチについて説明します。

はじめに

ワークロードは、意図した機能を正確に一貫して実行する必要があります。これを実現するには、回復性を考慮したアーキテクチャを設計する必要があります。回復性とは、インフラストラクチャやサービスの中断から復旧して、需要を満たすコンピューティングリソースを動的に獲得し、設定ミスや一時的なネットワーク問題などの中断の影響を緩和するワークロードの能力です。

災害対策 (DR) は、回復性戦略の重要な部分であり、災害の発生時にワークロードがどのように対応するかに関連します (災害はビジネスに深刻な悪影響を及ぼすイベントです)。この対応は、組織のビジネス目標に基づいている必要があります。ビジネス目標は、目標復旧時点 (RPO) と呼ばれるデータ損失を回避し、目標復旧時間 (RTO) と呼ばれるワークロードを使用できないダウンタイムを削減するためのワークロードの戦略を指定します。したがって、特定の 1 回限りの災害イベントに対する復旧目標 (RPO と RTO) を満たすために、クラウド内のワークロードの設計に回復性を実装する必要があります。このアプローチは、組織がビジネス継続性計画 (BCP) の一環としてビジネス継続性を維持するのに役立ちます。

このホワイトペーパーでは、ビジネスの災害対策目標を満たすアーキテクチャを AWS で計画、設計、実装する方法に焦点を当てています。ここで共有する情報は、CTO、アーキテクト、デベロッパー、運用チームのメンバーなど、テクノロジーの役割の担当者を対象としています。

災害対策と可用性

災害対策は、可用性と比較できます。可用性は、回復性戦略の別の重要な要素です。災害対策は 1 回限りのイベントに対する目標を測定するのに対し、可用性の目標は一定期間の平均値を測定することにあります。

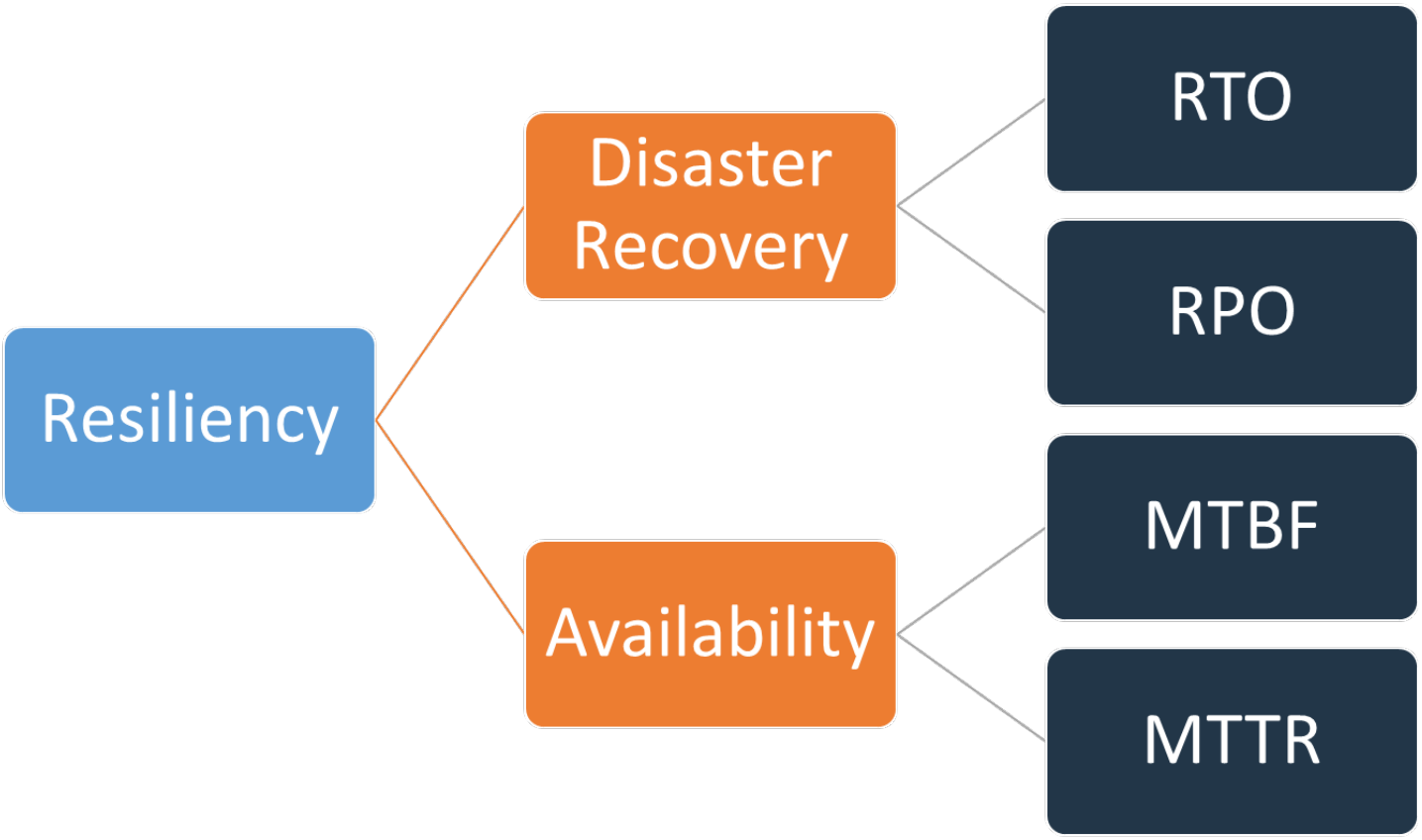

図1 - 回復性の目標

可用性は、MTBF と平均復旧時間 (MTTR) を使用して計算します。

$$Availability = \frac{Available\ for\ Use\ Time}{Total\ Time} = \frac{MTBF}{MTBF + MTTR}$$

このアプローチは、通常「ナイン」と呼ばれます。99.9% の可用性ターゲットは「スリーナイン」と呼ばれます。

ワークロードでは、時間ベースのアプローチを使用するよりも、成功および失敗したリクエストの数をカウントする方が簡単な場合があります。この場合、次の計算式を使用できます。

$$Availability = \frac{Successful\ Responses}{Valid\ Requests}$$

災害対策は災害イベントに重点を置き、可用性はコンポーネントの障害、ネットワーク問題、負荷のスパイクなど、より小規模な要素のより一般的な中断に重点を置いています。災害対策の目的はビジネス継続性であり、可用性は、目的のビジネス機能を実行するためにワークロードを利用できる時間を最大化することに関連しています。どちらも、回復性戦略の一部として組み込む必要があります。

回復性に関する責任共有モデル

回復性は、AWS とお客様との間での責任共有です。この共有モデルにおいて、災害対策と可用性が回復性の一環としてどのように機能するかを理解することが重要です。

AWS の責任「クラウドの回復性」

AWS は、AWS クラウド内で提供するすべてのサービスを実行するインフラストラクチャの回復性に責任があります。このインフラストラクチャは、AWS クラウドサービスを実行するハードウェア、ソフトウェア、ネットワーク、施設で構成されます。AWS は、これらの AWS クラウドサービスを利用可能にするために商業上合理的な努力を払い、サービスの可用性が AWS のサービスレベルアグリーメント (SLA) を満たすか上回ることを保証します。

AWS グローバルクラウドインフラストラクチャは、お客様が回復性の高いワークロードアーキテクチャを構築できるように設計されています。AWS リージョン間は完全に分離され、各リージョンは複数のアベイラビリティーゾーンで構成されています。アベイラビリティーゾーンは、インフラストラクチャの物理的に分離されたパーティションです。アベイラビリティーゾーンは、ワークロードの回復性に影響を与える可能性のある障害を隔離し、リージョン内の他のゾーンに影響を与えないようにします。ただし、同時に AWS リージョン内のすべてのアベイラビリティーゾーン間は、高スループットで低レイテンシーのネットワークを提供するフルリダンダントな専用メトロファイバーを介して、高帯域幅で低レイテンシーのネットワークで相互接続されています。ゾーン間のすべてのトラフィックは暗号化されます。ネットワークパフォーマンスは、ゾーン間の同期レプリケーションを十分に達成できます。アベイラビリティーゾーンは、アプリケーションをパーティション分割するプロセスを簡素化し、高可用性を確保します。

お客様の責任「クラウド内の回復性」

お客様の責任は、選択した AWS クラウドサービスによって決まります。これにより、回復性の責任の一環として実行する必要がある設定作業の量が決まります。例えば、Amazon Elastic Compute Cloud (Amazon EC2) などのサービスでは、すべての必要な回復性設定および管理タスクを実行する必要があります。Amazon EC2 インスタンスをデプロイするお客様は、複数の場所 (AWS アベイラビリティーゾーンなど) への EC2 インスタンスのデプロイ、自己修復の実装 (AWS Auto Scaling などのサービスを使用)、インスタンスにインストールされているアプリケーションに対する回復性の高いワークロードアーキテクチャのベストプラクティスの使用に責任があります。Amazon S3 や Amazon DynamoDB などのマネージドサービスについては、AWS がインフラストラクチャレイヤー、オペレーティングシステム、プラットフォームを運用し、お客様がエンドポイントにアクセス

してデータを保存および取得します。お客様は、バックアップ、バージョニング、およびレプリケーション戦略など、データの回復性を管理する責任を負います。

AWS リージョン内の複数のアベイラビリティーゾーンにワークロードをデプロイすることは、高可用性戦略の一部であり、1 つのアベイラビリティーゾーン内に問題を隔離することでワークロードを保護し、他のアベイラビリティーゾーンの冗長性を利用してリクエストの処理を継続します。マルチ AZ アーキテクチャも DR 戦略の一部であり、停電、落雷、竜巻、地震などの問題からワークロードを適切に隔離して保護するように設計されています。DR 戦略では、複数の AWS リージョンを利用することもできます。例えば、アクティブ/パッシブ設定の場合、アクティブなリージョンでリクエストを処理できなくなると、ワークロードのサービスはアクティブリージョンから DR 用リージョンにフェイルオーバーします。

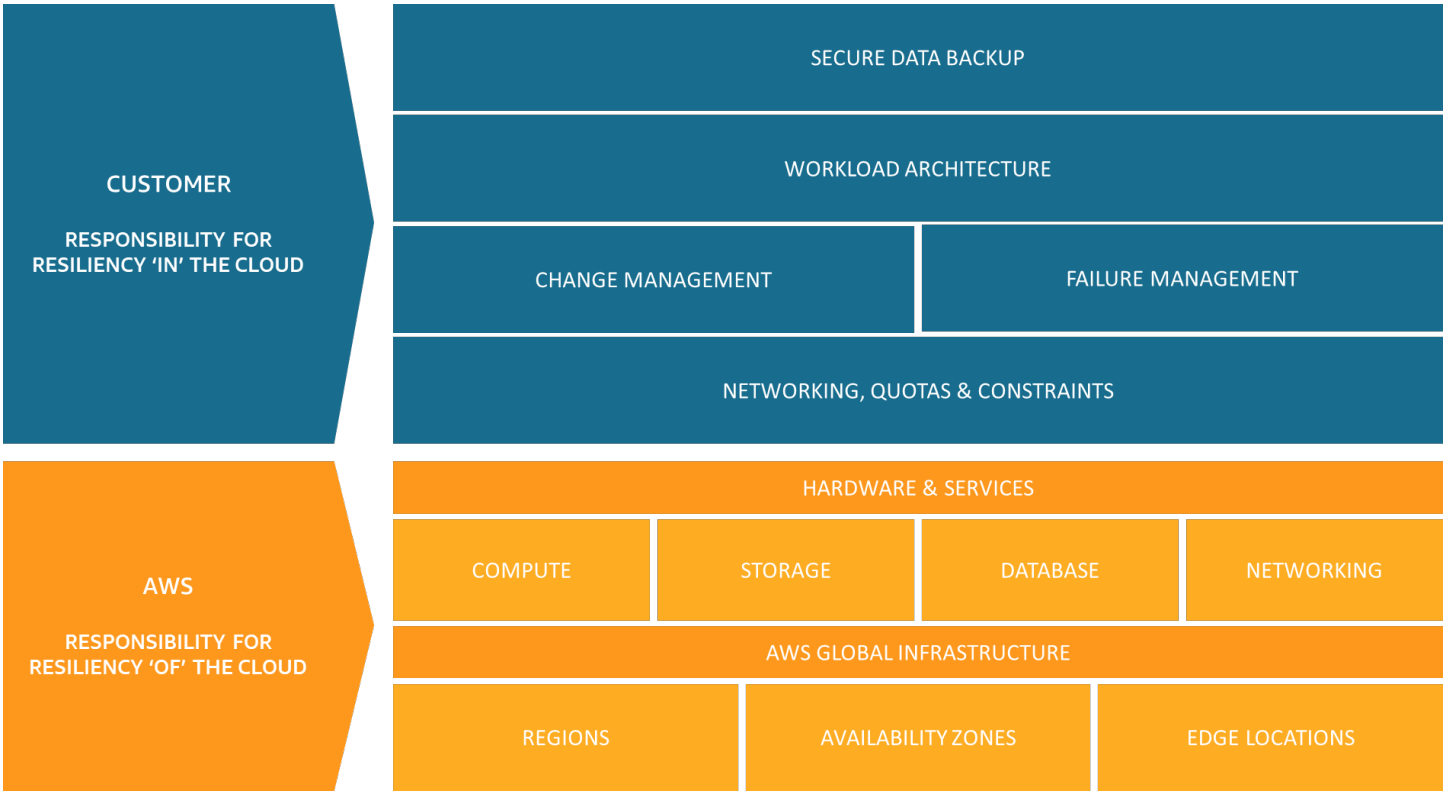

図 2 - 回復性は AWS とお客様との間での責任共有

災害とは

災害対策の計画策定では、次の 3 つの主要なカテゴリの災害に対する計画を評価します。

- 地震や洪水などの自然災害
- 停電やネットワーク接続などの技術的障害
- 不注意による設定ミス、不正アクセス/外部からのアクセス、改ざんなどの人的行為

これらの潜在的な各災害は、地方、地域、国、大陸、または地球レベルの地理的影響ももたらします。災害対策戦略を検討する場合は、災害の性質と地理的影響の両方が重要となります。例えば、地方の洪水によるデータセンターの停電問題を軽減するには、マルチ AZ 戦略を採用できます。各地方の問題は、複数のアベイラビリティーゾーンには影響を与えません。ただし、本番データに対する攻撃に対しては、別の AWS リージョン内のバックアップデータにフェイルオーバーする災害対策戦略を実行する必要があります。

高可用性は災害対策ではない

可用性と災害対策は、両方ともいくつかの同じベストプラクティス (障害のモニタリング、複数箇所へのデプロイ、自動フェイルオーバーなど) に依存しています。ただし、可用性はワークロードのコンポーネントに重点を置き、災害対策はワークロード全体の個別のコピーに重点を置いています。災害対策と可用性では目標が異なり、災害対策は災害に該当する大規模なイベントの発生から復旧するまでの時間を測定します。まず、ワークロードが可用性の目標を満たしていることを確認する必要があります。高可用性のアーキテクチャであれば、可用性に影響するイベントが発生した場合でも、お客様のニーズを満たすことができます。災害対策戦略では、可用性に対するアプローチとは異なるアプローチが必要であり、複数の場所に個別のシステムをデプロイし、必要に応じてワークロード全体をフェイルオーバーできることが重要です。

災害対策計画では、ワークロードの可用性を考慮する必要があります。可用性は、使用するアプローチに影響するためです。1 つのアベイラビリティーゾーン内の 1 つの Amazon EC2 インスタンスで実行されるワークロードには、高可用性がありません。地域で洪水が発生してアベイラビリティーゾーンが影響を受けた場合、このシナリオでは別の AZ にフェイルオーバーして DR の目標を達成する必要があります。このシナリオを、マルチサイト アクティブ/アクティブにデプロイした高可用性ワークロードと比較してください。マルチサイト アクティブ/アクティブでは、ワークロードを複数のアクティブなリージョンにデプロイし、すべてのリージョンで本番トラフィックを処理します。この場合、大規模な災害によってリージョン全体が破壊されたとしても、すべてのトラフィックを残りのリージョンにルーティングすることで DR 戦略が達成されます。

データへのアプローチ方法も、可用性と災害対策では異なります。高可用性を実現するには、別のサイトに継続的にレプリケートするストレージソリューション (マルチサイト アクティブ/アクティブワークロードなど) を検討します。プライマリストレージデバイス上で 1 つまたは複数のファイルが削除または破損すると、これらの破壊的な変更がセカンダリストレージデバイスにレプリケートされる場合があります。このシナリオでは、高可用性にもかかわらず、データの削除や破損に伴うフェイルオーバー機能が損なわれます。代わりに、DR 戦略の一環としてポイントインタイムバックアップも必要です。

ビジネス継続性計画 (BCP)

災害対策計画は、組織のビジネス継続性計画 (BCP) の一部として含める必要があります。独立したドキュメントにはしません。ワークロードを復元する果敢な災害対策目標を掲げても、そのワークロード以外のビジネス要素が受けた災害のせいでワークロードのビジネス目標が達成できなければ、何の意味もありません。例えば、地震のせいで e コマースアプリケーションで販売した製品を配送できなくなる場合があります。この場合、BCP で対処すべきニーズは配送であり、効果的な DR でワークロードの機能を維持しても配送できなければ意味がありません。DR 戦略は、ビジネス要件、優先順位、コンテキストに基づいて策定する必要があります。

ビジネスインパクト分析とリスク評価

ビジネスインパクト分析では、ワークロードの中断がビジネスに及ぼす影響を定量化する必要があります。ワークロードの使用不能が社内外のお客様に与える影響と、ビジネスに及ぼす影響を見極める必要があります。この分析は、ワークロードをどれだけ早く利用可能にする必要があるか、およびデータ損失をどれだけ許容できるかを判断するのに役立ちます。ただし、注意点として、復旧目標は単独で考慮しないことが重要です。ワークロードの災害対策をビジネス価値として提示するには、中断の可能性と復旧のコストを重要な要因として考慮に入れる必要があります。

ビジネスへの影響は時間に依存する場合があります。この点を災害対策計画に組み込むことを検討してください。例えば、供与システムの中断は、給与の支払い直前であればビジネスに非常に大きな影響を与える可能性がありますが、給与の支払い直後であれば影響は少ない可能性があります。

災害の種類と地理的影響のリスク評価と、ワークロードの技術的な実装の概要によって、災害の種類ごとに発生する中断の可能性が決まります。

非常に重要なワークロードに対しては、ビジネスへの影響を最小限に抑えるために、複数のリージョンにわたって継続的なバックアップを行い、高可用性を確保することを検討します。重要度の低いワークロードに対しては、災害対策を一切講じないことも有効な戦略となります。また、災害シナリオによっては、災害発生の可能性が低いと確信できる場合、災害対策戦略を持たないことも有効です。AWS リージョン内のアベイラビリティーゾーンは、相互に十分な距離を取って、慎重に場所が計画されているため、ほとんどの一般的な災害は 1 つのゾーンには影響しても、他のゾーンには影響しないはずです。したがって、AWS リージョン内のマルチ AZ アーキテクチャは、リスク軽減のニーズを既に満たしている可能性があります。

災害対策オプションのコストを評価し、災害対策戦略がビジネスへの影響とリスクを反映した適切なレベルのビジネス価値を提供することを確認する必要があります。

これらすべての情報に基づいて、さまざまな災害シナリオの脅威、リスク、影響、コストを、関連する復旧オプションとともに文書化できます。これらの情報を使用して、各ワークロードの復旧目標を決定する必要があります。

復旧目標 (RTO と RPO)

災害対策 (DR) 戦略を作成する場合、最も一般的な方法として、組織は目標復旧時間 (RTO) と目標復旧時点 (RPO) を計画します。

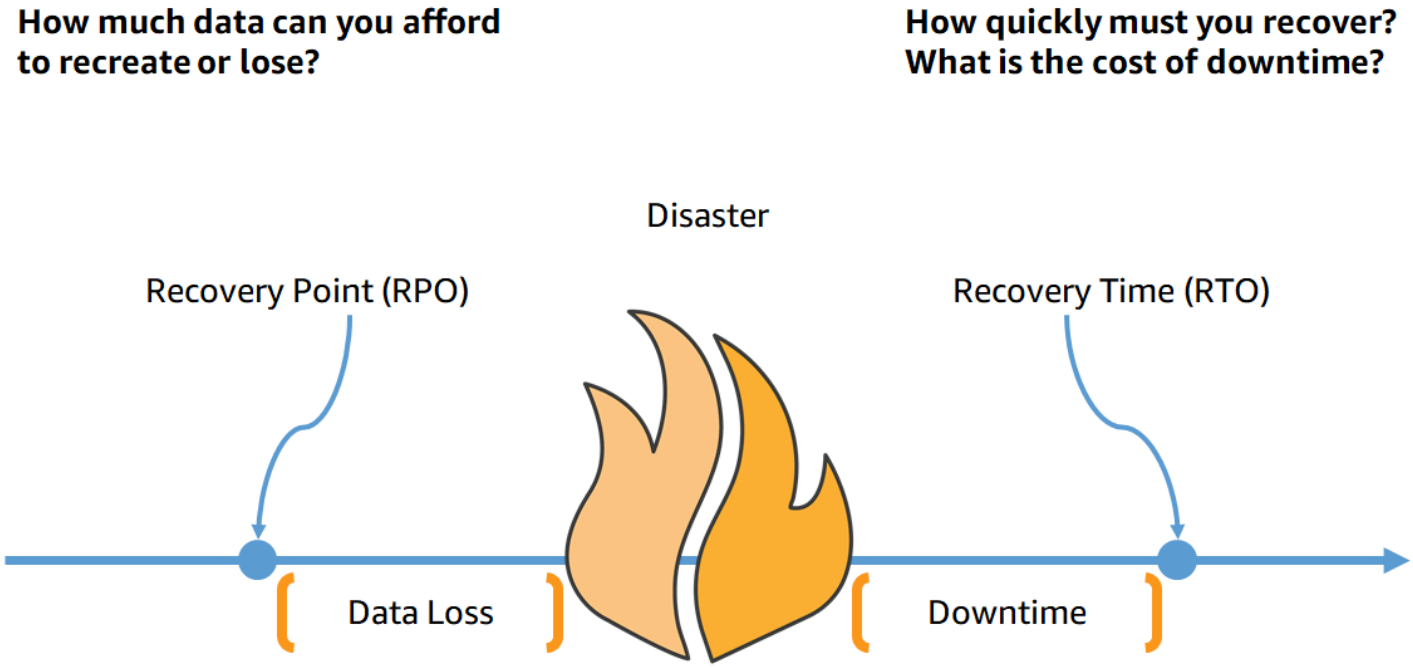

図 3 - 復旧目標

目標復旧時間 (RTO) は、サービスの中断からサービスの復元までの最大許容遅延です。この目標は、サービスが利用できない時間枠としてどの程度を許容するかを決定するものであり、組織が定義します。

このホワイトペーパーでは、主に「バックアップと復元」、「パイロットライト」、「ウォームスタンバイ」、「マルチサイト アクティブ/アクティブ」の 4 つの DR 戦略について説明します (「クラウド内の災害対策オプション」を参照してください)。次の図で、ビジネスは最大許容 RTO と、サービス復旧戦略のコストの限度額を決定しています。ビジネス目標を考慮すると、DR 戦略のパイロットライトまたはウォームスタンバイは RTO とコスト基準の両方を満たします。

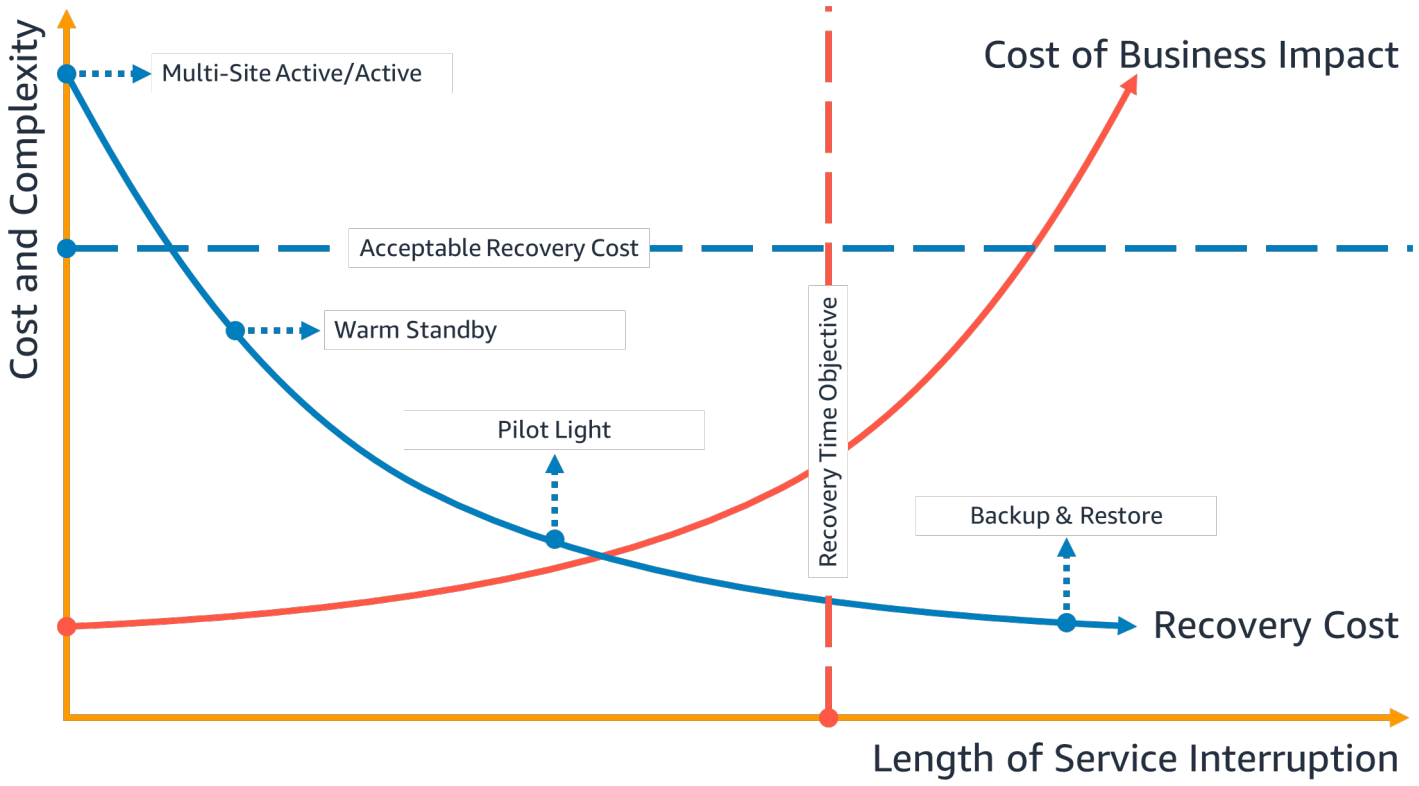

図 4 - 目標復旧時間

目標復旧時点 (RPO) は、最後のデータ復旧時点からの最大許容時間です。この目標は、最後の復旧時点からサービスの中断までの間にどの程度のデータ損失を許容するかを決定するものであり、組織が定義します。

次の図で、ビジネスは最大許容 RPO と、データ復旧戦略のコストの限度額を決定しています。4 つの DR 戦略のうち、パイロットライトまたはウォームスタンバイの DR 戦略は RPO とコストの両方の基準を満たしています。

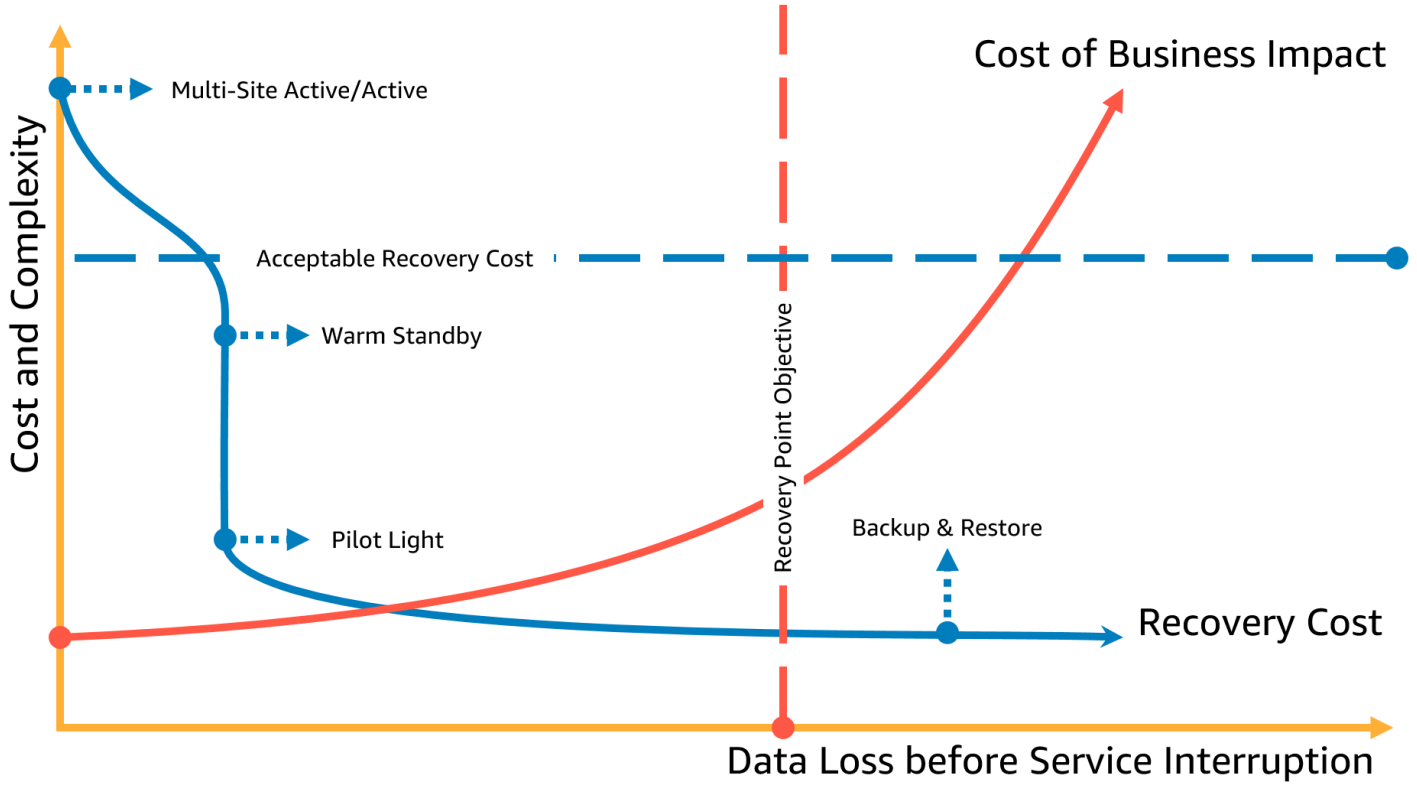

図 5 - 目標復旧時点

> **Note**
>
> 復旧のコストが障害や損失のコストよりも高い場合は、規制要件などの二次的な要因がない限り、復旧オプションを設定しません。

クラウド内の災害対策は異なる

災害対策戦略は、技術革新とともに進化します。オンプレミスの災害対策計画では、テープを物理的に移動したり、データを別のサイトにレプリケートしたりすることがあります。組織が AWS で DR 目標を達成するには、以前の災害対策戦略に伴うビジネスへの影響、リスク、コストを再評価する必要があります。AWS クラウド内の災害対策には、従来の環境と比べて、次のような利点があります。

- 複雑さを軽減して、災害からの復旧を迅速化する
- シンプルで再現性のあるテストにより、より簡単かつ頻繁にテストできる
- 管理オーバーヘッドを削減して運用上の負担を減らす
- 自動化してエラーの可能性を減らし、復旧時間を短縮する機会がある

AWS では、物理バックアップデータセンターの固定資本費を、クラウド内の適正規模の環境の変動運用費に転換できるため、コストを大幅に削減できます。

多くの組織にとって、オンプレミスの災害対策では、データセンター内のワークロードが中断するリスクがあり、セカンダリデータセンターにバックアップまたはレプリケートしたデータを復旧する必要がありました。組織は、AWS にワークロードをデプロイすることで、Well-Architected ワークロードを実装し、AWS グローバルクラウドインフラストラクチャの設計を活用できるため、そのような中断の影響を軽減できます。クラウド内で信頼性、安全性、効率性、コスト効率に優れたワークロードを設計および運用するためのアーキテクチャ上のベストプラクティスの詳細については、「AWS Well-Architected Framework - 信頼性の柱」ホワイトペーパーを参照してください。

ワークロードが AWS にある場合、データセンターの接続 (データセンターへのアクセスは除く)、電力、空調、防火、ハードウェアについて心配する必要はありません。これらはすべて自動的に管理され、障害から隔離された複数のアベイラビリティーゾーン (それぞれが 1 つ以上の個別のデータセンターで構成) にアクセスできます。

単一の AWS リージョン

1 つの物理データセンターの中断または損失に基づく災害イベントの場合、単一の AWS リージョン内の複数のアベイラビリティーゾーンに高可用性のワークロードを実装することで、自然災害や技術災害を軽減し、データの損失につながるエラーや不正行為などの人的脅威のリスクを軽減できます。各 AWS リージョンは複数のアベイラビリティーゾーンで構成され、各アベイラビリティーゾーンは他のゾーンで発生した障害から隔離されています。各アベイラビリティーゾーンは、複数の物理

データセンターで構成されています。影響の大きい問題をより適切に切り分けて高可用性を実現するには、同じリージョン内の複数のゾーンにワークロードを分割できます。アベイラビリティーゾーンは、物理的な冗長性を確保し、耐障害性を提供するように設計されているため、停電、インターネットのダウンタイム、洪水、その他の自然災害が発生した場合でも、中断のないパフォーマンスが可能です。AWS がこれを行う方法については、「AWS グローバルクラウドインフラストラクチャ」を参照してください。

単一の AWS リージョンの複数のアベイラビリティーゾーンにデプロイすることで、1 つ (または複数) のデータセンターの障害からワークロードをより適切に保護できます。単一リージョンのデプロイをさらに確実にするために、データと設定 (インフラストラクチャ定義を含む) を別のリージョンにバックアップできます。この戦略により、災害対策計画の範囲を縮小してデータのバックアップと復元だけを含めることができます。別の AWS リージョンにバックアップしてマルチリージョンの回復性を活用することは、次のセクションで説明する他のマルチリージョンオプションと比べて簡単で安価です。例えば、 Amazon Simple Storage Service (Amazon S3) にバックアップすると、データをすぐに取り出すことができます。ただし、データの一部に対する DR 戦略で、取り出し時間の要件が (数分から数時間に) 緩和されている場合は、Amazon S3 Glacier または Amazon S3 Glacier Deep Archive を使用することで、バックアップと復旧戦略のコストを大幅に削減できます。

ワークロードによっては、データ所在地に関する規制要件が伴う場合があります。これが AWS リージョンが現在 1 つしかない地域のワークロードに当てはまる場合は、上記で説明したように高可用性を実現するマルチ AZ ワークロードの設計に加えて、そのリージョン内の AZ を個別の口ケーションとして使用することもできます。これにより、そのリージョン内のワークロードに適用されるデータ所在地要件に対応できます。以下のセクションで説明する DR 戦略は、複数の AWS リージョンを使用していますが、リージョンの代わりにアベイラビリティーゾーンを使用して実装することもできます。

複数の AWS リージョン

データセンター間が十分に離れているにもかかわらず、複数のデータセンターの機能が停止するような災害イベントに備えて、AWS 内のリージョン全体に影響する自然災害や技術災害を軽減するための災害対策オプションを検討する必要があります。以下のセクションで説明するすべてのオプションは、そのような災害から保護するためのマルチリージョンアーキテクチャとして実装できます。

クラウド内での災害対策オプション

AWS 内で利用できる災害対策戦略は、コストが低くて複雑さが少ないバックアップ作成から、複数のアクティブなリージョンを使用したより複雑な戦略まで、4 つのアプローチに大別できます。災害対策戦略を定期的にテストして、必要なときに自信を持って実行できるようにすることが重要です。

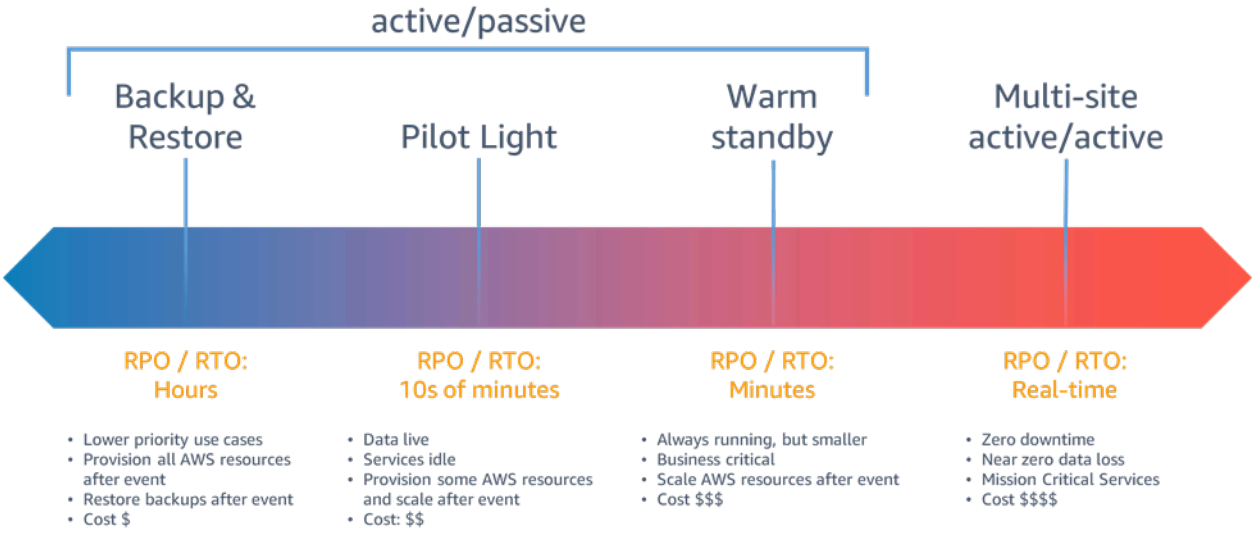

図 6 - 災害対策戦略

高可用性の Well-Architected ワークロードに対する災害イベントが、1 つの物理データセンターの中断や喪失に伴うものである場合、災害対策として必要なのはバックアップと復元のアプローチのみである場合があります。災害の定義が、物理データセンターの中断や喪失を超えてリージョンの災害にまで及ぶ場合、または定義を必要とする規制要件の適用対象である場合は、パイロットライト、ウォームスタンバイ、またはマルチサイト アクティブ/アクティブを検討する必要があります。

バックアップと復元

バックアップと復元は、データの損失や破損を軽減するために適したアプローチです。このアプローチは、他の AWS リージョンにデータをレプリケートしてリージョンの災害を軽減したり、1 つのアベイラビリティーゾーンにデプロイしたワークロードの冗長性の欠如を軽減したりするためにも使用できます。データに加えて、インフラストラクチャ、設定、アプリケーションコードを復旧先のリージョンに再デプロイする必要があります。インフラストラクチャを エラーなく迅速に再デプロイするには、AWS CloudFormation や AWS Cloud Development Kit (AWS CDK) などのサービスを通じて常に Infrastructure as Code (IaC) を使用してデプロイする必要があります。IaC を使用しないと、復旧先のリージョンにワークロードを復元することが複雑になり、復旧時間が長くなって

RTO を超えるおそれがあります。ユーザーデータに加えて、Amazon EC2 インスタンスの作成に使用する Amazon マシンイメージ (AMI) など、コードと設定も必ずバックアップしてください。AWS CodePipeline を使用すると、アプリケーションコードと設定の再デプロイを自動化できます。

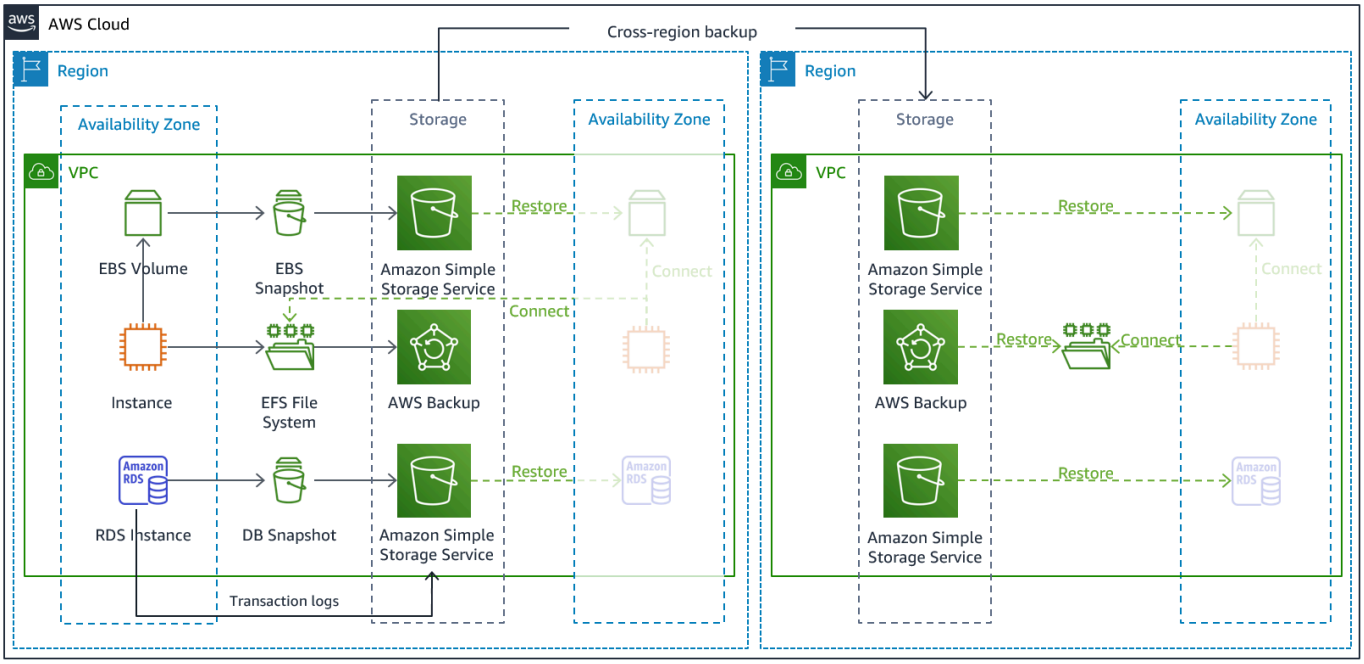

図 7 - バックアップと復元のアーキテクチャ

AWS のサービス

ワークロードのデータに対しては、バックアップ戦略を定期的または継続的に実行する必要があります。バックアップを実行する頻度によって、達成できる復旧時点が決まります (RPO を満たすように調整する必要があります)。バックアップは、バックアップを作成した時点に復元する方法も提供する必要があります。ポイントインタイムリカバリによるバックアップは、以下のサービスとリソースを通じて利用できます。

- Amazon Elastic Block Store (Amazon EBS) スナップショット
- Amazon DynamoDB バックアップ
- Amazon RDS スナップショット
- Amazon Aurora DB スナップショット
- Amazon EFS バックアップ (AWS Backup の使用時)
- Amazon Redshift スナップショット
- Amazon Neptune スナップショット

Amazon Simple Storage Service (Amazon S3) では、Amazon S3 クロスリージョンレプリケーション (CRR) を使用して、オブジェクトを継続的に DR 用リージョンの S3 バケットに非同期的にコピーするとともに、保存したオブジェクトのバージョニングを提供して復元ポイントを選択できます。データの継続的レプリケーションには、データのバックアップ時間が最短 (ゼロに近い) という利点がありますが、データの破損や悪意のある攻撃 (不正なデータ削除など) などの災害イベントから保護できない場合があります。ポイントインタイムバックアップもできません。継続的レプリケーションについては、「パイロットライト向けの AWS のサービス」セクションで説明しています。

AWS Backup は、以下のサービスとリソースに対して AWS のバックアップ機能を設定、スケジュール、モニタリングするための一元化された場所を提供します。

- Amazon Elastic Block Store (Amazon EBS) ボリューム
- Amazon EC2 インスタンス
- Amazon Relational Database Service (Amazon RDS) データベース (Amazon Aurora データベースを含む)
- Amazon DynamoDB テーブル
- Amazon Elastic File System (Amazon EFS) ファイルシステム
- AWS Storage Gateway ボリューム
- Amazon FSx for Windows ファイルサーバーおよび Amazon FSx for Lustre

AWS Backup は、リージョン間でのバックアップのコピー (災害対策リージョンへのコピーなど) をサポートしています。

Amazon S3 データの追加の災害対策戦略として、S3 オブジェクトのバージョニングを有効にします。オブジェクトのバージョニングは、削除や変更のアクションの前に元のバージョンを保持することで、アクションの結果から S3 内のデータを保護します。オブジェクトのバージョニングは、人為的なミスに伴う災害への有効な軽減策となります。S3 レプリケーションを使用してデータを DR 用リージョンにバックアップしている場合、デフォルトでは、レプリケート元バケットでオブジェクトが削除されると、Amazon S3 はレプリケート元バケットにのみ削除マーカーを追加します。このアプローチは、DR 用リージョンのデータをレプリケート元リージョンでの悪意のある削除から保護します。

データに加えて、ワークロードの再デプロイと目標復旧時間 (RTO) の達成に必要な設定とインフラストラクチャもバックアップする必要があります。AWS CloudFormation では、Infrastructure as Code (IaC) を提供し、ユーザーがワークロード内のすべての AWS リソースを定義して、複数の AWS アカウントや AWS リージョンに確実にデプロイおよび再デプロイできるようにします。ワー

クロードで使用する Amazon EC2 インスタンスを Amazon マシンイメージ (AMI) としてバックアップできます。AMI は、インスタンスのルートボリュームと、インスタンスにアタッチされたその他の EBS ボリュームのスナップショットから作成します。この AMI を使用して、復元したバージョンの EC2 インスタンスを起動できます。AMI のコピーはリージョン内またはリージョン間で行うことができます。または、AWS Backup を使用して、バックアップをアカウント間でコピーしたり、他の AWS リージョンにコピーしたりできます。クロスアカウントバックアップ機能は、内部関係者による脅威やアカウントの侵害などの災害イベントからの保護に役立ちます。AWS Backup は、EC2 バックアップに他の機能も追加します。インスタンスの個々の EBS ボリュームに加えて、AWS Backup は、インスタンスタイプ、設定済みの仮想プライベートクラウド (VPC)、セキュリティグループ、IAM ロール、モニタリング設定、タグなどのメタデータも保存および追跡します。ただし、この追加メタデータは、EC2 バックアップを同じ AWS リージョンに復元する場合にのみ使用します。

災害対策用リージョンにバックアップとして保存したデータは、フェイルオーバー時に復元する必要があります。AWS Backup には復元機能がありますが、現在、スケジュールされた復元や自動復元は利用できません。AWS SDK を使用して AWS Backup の API を呼び出すと、DR 用リージョンへの自動復元を実装できます。これを定期的に繰り返すジョブとして設定したり、バックアップが完了するたびに復元をトリガーしたりできます。次の図は、Amazon Simple Notification Service (Amazon SNS) と AWS Lambda を使用した自動復元の例を示しています。

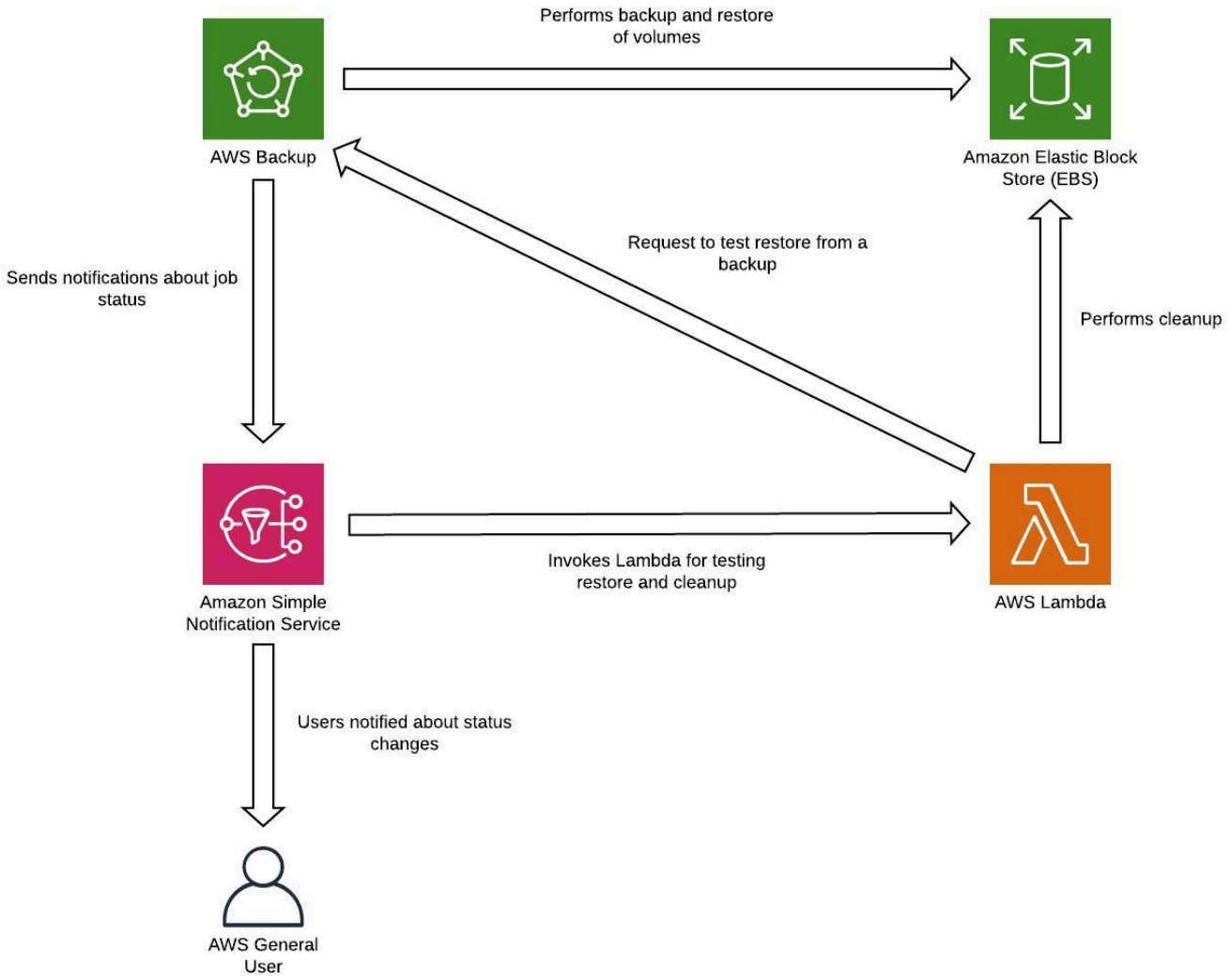

図 8 - バックアップの復元とテスト

> **Note**
>
> バックアップ戦略には、バックアップのテストを含める必要があります。詳細については、「災害対策のテスト」セクションを参照してください。実装の実践デモについては、「AWS Well-Architected ラボ: データのバックアップと復元のテスト」を参照してください。

パイロットライト

パイロットライトアプローチでは、あるリージョンから別のリージョンにデータをレプリケートし、コアワークロードインフラストラクチャのコピーをプロビジョニングします。データベースやオブ

ジェクトストレージなど、データのレプリケーションとバックアップをサポートするために必要なリソースは常に稼働しています。アプリケーションサーバーなどの他の要素は、アプリケーションコードや設定とともにロードされますが、オフに切り替えられ、テスト時または災害対策フェイルオーバーの起動時にのみ使用されます。バックアップと復元のアプローチとは異なり、コアインフラストラクチャは常に利用可能であり、アプリケーションサーバーをオンに切り替えてスケールアウトすることで、フルスケールの本番環境を迅速にプロビジョニングできます。

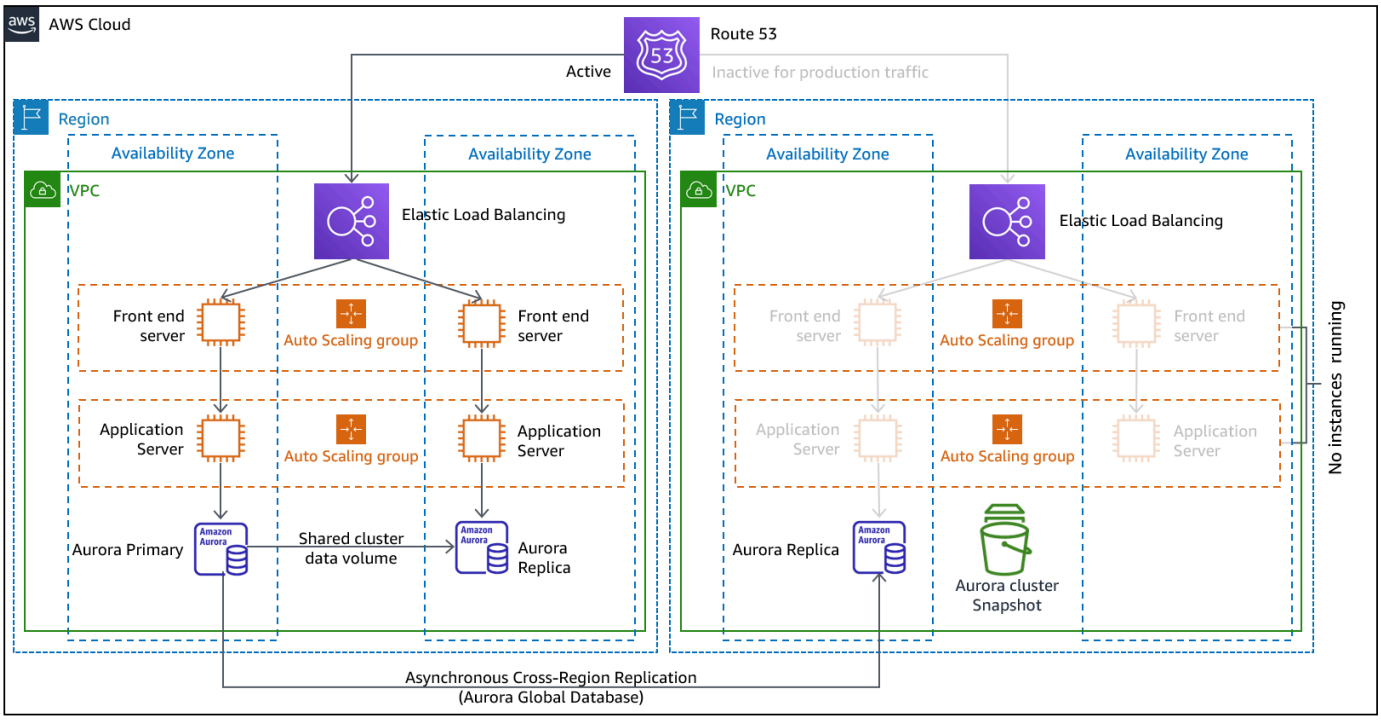

図 9 - パイロットライトのアーキテクチャ

パイロットライトアプローチでは、アクティブなリソースを最小限に抑えて災害対策の継続的なコストを最小化します。コアインフラストラクチャの要件がすべて揃っているため、災害発生時の復旧は簡素化されます。この復旧オプションでは、デプロイ方法を変更する必要があります。各リージョンに対してコアインフラストラクチャの変更を行い、ワークロード (設定、コード) の変更を各リージョンに同時にデプロイする必要があります。このステップを簡素化するには、デプロイを自動化し、Infrastructure as Code (IaC) を使用して複数のアカウントやリージョンにインフラストラクチャをデプロイします (インフラストラクチャ全体をプライマリリージョンにデプロイし、スケールダウン/オフに切り替えたインフラストラクチャを DR 用リージョンにデプロイします)。最高レベルのリソースとセキュリティの分離を実現するために、リージョンごとに異なるアカウントを使用することをお勧めします (認証情報の侵害が災害対策計画の一部である場合も)。

このアプローチでは、データ災害も軽減する必要があります。継続的なデータレプリケーションは、いくつかの種類の災害からユーザーを保護しますが、保存データのバージョニングやポイントインタ

イムリカバリのオプションも戦略に含めていない限り、データの破損や破壊からユーザーを保護することはできません。レプリケートしたデータを災害リージョンでバックアップして、同じリージョンにポイントインタイムバックアップを作成できます。

AWS のサービス

「バックアップと復元」セクションで説明した AWS のサービスを使用してポイントインタイムバックアップを作成することに加えて、パイロットライト戦略では次のサービスも検討してください。

パイロットライトでは、DR 用リージョン内のライブデータベースやデータストアへの継続的なデータレプリケーションが、RPO を低く抑えるための最善のアプローチです (前述のポイントインタイムバックアップに加えて使用する場合)。AWS では、以下のサービスとリソースを使用して、データをクロスリージョンで継続して非同期的にレプリケートできます。

- Amazon Simple Storage Service (Amazon S3) レプリケーション
- Amazon RDS リードレプリカ
- Amazon Aurora Global Database
- Amazon DynamoDB グローバルテーブル

継続的レプリケーションでは、データのバージョンを DR 用リージョンでほぼ即座に利用できます。実際のレプリケーション時間は、S3 オブジェクト用の S3 レプリケーション時間制御 (S3 RTC) や、Amazon Aurora Global Database の管理機能などのサービス機能を使用してモニタリングできます。

フェイルオーバーして災害対策用リージョンから読み取り/書き込みワークロードを実行する場合は、RDS リードレプリカをプライマリインスタンスに昇格させる必要があります。Aurora 以外の DB インスタンスの昇格は、完了するまでに数分かかり、昇格プロセスの一部として再起動が必要です。クロスリージョンレプリケーション (CRR) と RDS によるフェイルオーバーでは、Amazon Aurora Global Database を使用するといくつかの利点があります。グローバルデータベースは、専用のインフラストラクチャを使用し、アプリケーションでデータベースを全面的に利用できるようにします。また、セカンダリリージョンへのレプリケーションに伴うレイテンシーは、一般的に 1 秒未満 (AWS リージョン内では 100 ミリ秒未満) です。Amazon Aurora Global Database を使用すると、プライマリリージョンでパフォーマンスの低下や停止が発生し、リージョン全体が停止した場合でも、1 分以内にセカンダリリージョンの 1 つを昇格させて読み取り/書き込みの責任を引き継ぐことができます。昇格は自動化することが可能であり、再起動は不要です。

DR 用リージョンには、コアワークロードインフラストラクチャのスケールダウンしたバージョン (リソースの数が少ないか、サイズが小さい) をデプロイする必要があります。AWS CloudFormation

を使用すると、インフラストラクチャを定義し、これを AWS アカウントや AWS リージョン全体に一貫してデプロイできます。AWS CloudFormation では、事前定義された擬似パラメータを使用して、デプロイ先の AWS アカウントや AWS リージョンを特定します。したがって、CloudFormation テンプレートに条件ロジックを実装して、スケールダウンしたバージョンのインフラストラクチャのみを RD 用リージョンにデプロイできます。EC2 インスタンスのデプロイでは、Amazon マシンイメージ (AMI) がハードウェア設定やインストール済みソフトウェアなどの情報を提供します。必要な AMI を作成する Image Builder パイプラインを実装し、これらをプライマリリージョンとバックアップリージョンの両方にコピーできます。これにより、これらのゴールデン AMI は、災害発生時にワークロードを新しいリージョンに再デプロイまたはスケールアウトするための万全の備えを提供します。Amazon EC2 インスタンスは、スケールダウンした設定 (プライマリリージョンよりも少ないインスタンス数) でデプロイされます。休止状態を使用すると、EC2 インスタンスを停止状態にできます。この場合、EC2 の費用は発生せず、使用したストレージに対してのみ支払います。EC2 インスタンスを起動するには、AWS コマンドラインインターフェイス (CLI) または AWS SDK を使用してスクリプトを作成できます。インフラストラクチャをスケールアウトして本番トラフィックをサポートするには、「ウォームスタンバイ」セクションの「AWS Auto Scaling」を参照してください。

パイロットライトなどのアクティブ/スタンバイ設定では、すべてのトラフィックの送信先は最初にプライマリリージョンであり、プライマリリージョンが使用できなくなると、送信先が災害対策用リージョンに切り替わります。AWS のサービスの使用を検討する場合、2 つのトラフィック管理オプションがあります。最初のオプションは、Amazon Route 53 を使用することです。Amazon Route 53 を使用すると、1 つ以上の AWS リージョンにある複数の IP エンドポイントを 1 つの Route 53 ドメイン名に関連付けることができます。次に、そのドメイン名の下にある適切なエンドポイントにトラフィックをルーティングできます。Amazon Route 53 ヘルスチェックは、これらのエンドポイントをモニタリングします。これらのヘルスチェックを使用すると、トラフィックが正常なエンドポイントに確実に送信されるように DNS フェイルオーバーを設定できます。

2番目のオプションは、AWS Global Accelerator を使用することです。AnyCast IP を使用すると、1 つ以上の AWS リージョンにある複数のエンドポイントを同じ静的 IP アドレスに関連付けることができます。AWS Global Accelerator は、そのアドレスに関連付けられた適切なエンドポイントにトラフィックをルーティングします。Global Accelerator ヘルスチェックは、エンドポイントをモニタリングします。AWS Global Accelerator は、これらのヘルスチェックを使用して、アプリケーションの正常性を自動的にチェックし、ユーザートラフィックを正常なアプリケーションエンドポイントにのみルーティングします。Global Accelerator は、広範な AWS エッジネットワークを利用して、できるだけ早く AWS ネットワークバックボーンにトラフィックを送信するため、アプリケーションエンドポイントへのレイテンシーが短縮されます。また、Global Accelerator は DNS システム (Route 53 など) で発生する可能性のあるキャッシュの問題も回避します。

CloudEndure Disaster Recovery

CloudEndure Disaster Recovery は、AWS Marketplace から利用でき、基盤となるサーバーのブロックレベルレプリケーションを使用して、サーバーがホストするアプリケーションとサーバーがホストするデータベースを任意のソースから AWS に継続的にレプリケートします。CloudEndure Disaster Recovery により、AWS クラウドをオンプレミスのワークロードとその環境の災害対策用リージョンとして使用できます。また、AWS がホストするワークロードが EC2 (RDS ではない) でホストされているアプリケーションとデータベースのみで構成されている場合、これらのワークロードの災害対策にも使用できます。CloudEndure Disaster Recovery はパイロットライト戦略を使用し、ステージングエリアとして使用されている Amazon Virtual Private Cloud (Amazon VPC) に、データのコピーとオフに切り替えたリソースのコピーを保持します。フェイルオーバーイベントがトリガーされると、ステージングされたリソースを使用して、復旧場所として使用されているターゲット Amazon VPC にフル容量のデプロイが自動的に作成されます。

図 10 - CloudEndure Disaster Recovery アーキテクチャ

ウォームスタンバイ

ウォームスタンバイアプローチでは、本番環境のスケールダウンしたバージョンではあるが、完全な機能を備えたコピーを別のリージョンに確保します。このアプローチは、パイロットライトの拡張であり、別のリージョンでワークロードが常に稼動しているため、復旧までの時間が短縮されます。また、このアプローチでは、より簡単にテストを実行したり、継続的なテストを実装したりできるため、災害から復旧する能力に自信を深めることができます。

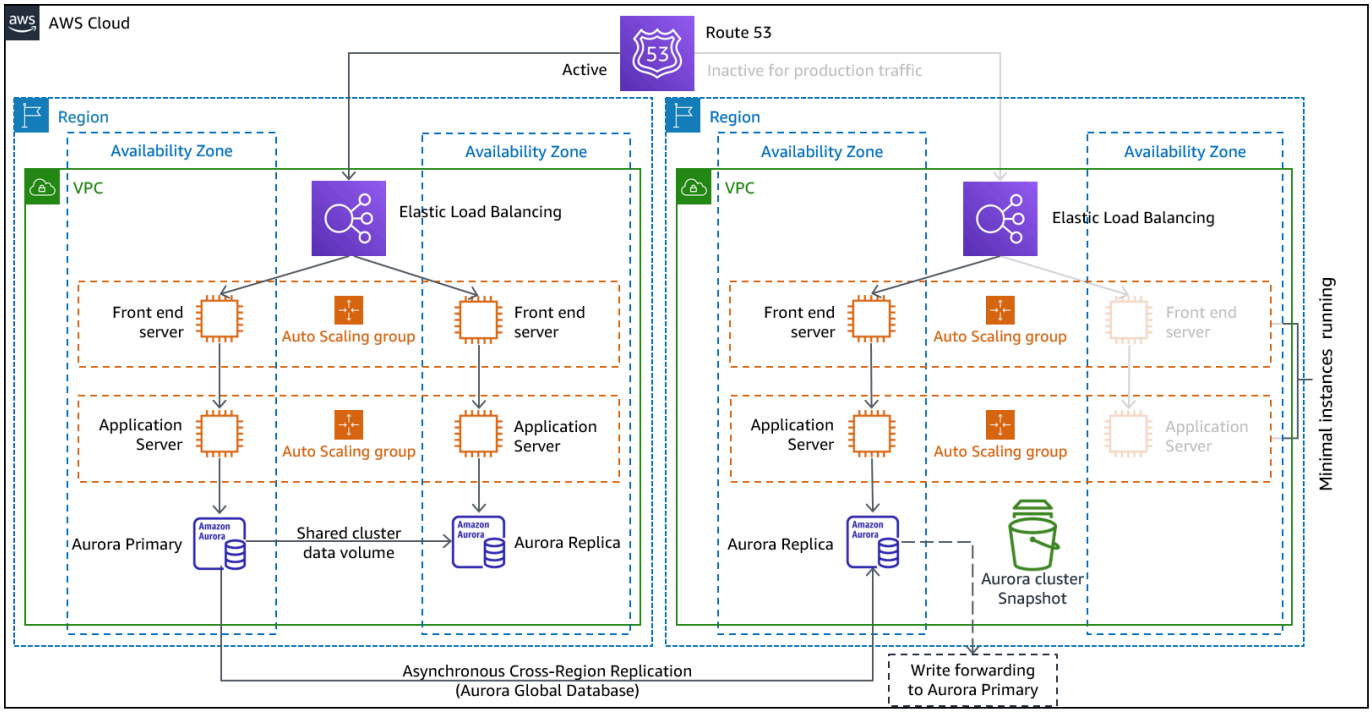

図 11 - ウォームスタンバイアーキテクチャ

注意: パイロットライトとウォームスタンバイの違いは理解しにくいかもしれません。どちらの場合も、プライマリリージョンのアセットが DR 用リージョンの環境にコピーされます。両者の違いは、パイロットライトは最初に追加のアクションを実行しないと要求を処理できないのに対し、ウォームスタンバイは (縮小した容量レベルで) トラフィックを即座に処理できる点です。パイロットライトアプローチでは、サーバーの「起動」、場合によっては追加 (コア以外) のインフラストラクチャのデプロイ、スケールアップが必要ですが、ウォームスタンバイでは、スケールアップのみが必要です (すべてはデプロイ済みで、既に実行されています)。RTO と RPO のニーズに応じて、これらのアプローチのいずれかを選択してください。

AWS のサービス

ウォームスタンバイでも、「バックアップと復元」と「パイロットライト」で取り上げたすべての AWS のサービスを、データのバックアップ、データのレプリケーション、アクティブ/スタンバイのトラフィックルーティング、EC2 インスタンスを含むインフラストラクチャのデプロイに使用します。

AWS Auto Scaling は、AWS リージョン内の Amazon EC2 インスタンス、Amazon ECS タスク、Amazon DynamoDB スループット、Amazon Aurora レプリカなどのリソースをスケールするために使用します。Amazon EC2 Auto Scaling は、AWS リージョン内のアベイラビリティーゾーン全

体で EC2 インスタンスのデプロイをスケールし、そのリージョン内での回復性を提供します。Auto Scaling を使用して、パイロットライト戦略またはウォームスタンバイ戦略の一環として、DR 用リージョンを実稼働能力にスケールアウトします。例えば、EC2 の場合は、Auto Scaling グループで目的の容量設定を増やします。この設定を手動で調整するには AWS Management Console を使用します。自動で調整するには、AWS SDK を使用します。または、新しい目的の容量値を使用し、AWS CloudFormation テンプレートを再デプロイして調整します。AWS CloudFormation のパラメータを使用すると、CloudFormation テンプレートをより簡単に再デプロイできます。実稼働容量へのスケールアップを制限しないように、DR 用リージョンのサービスクォータが十分に高く設定されていることを確認してください。

マルチサイト アクティブ/アクティブ

マルチサイト アクティブ/アクティブまたはホットスタンバイ アクティブ/パッシブ戦略の一環として、ワークロードを複数のリージョンで同時に実行できます。マルチサイト アクティブ/アクティブでは、デプロイ先のすべてのリージョンからのトラフィックを処理します。一方、ホットスタンバイでは、1 つのリージョンからのトラフィックのみを処理し、他のリージョンは災害対策専用として使用します。マルチサイト アクティブ/アクティブアプローチの場合、ユーザーはデプロイ先のすべてのリージョンでワークロードにアクセスできます。これは、災害対策として最も複雑でコストのかかるアプローチですが、適切なテクノロジーを選択して実装することで、ほとんどの災害で復旧時間をほぼゼロにまで短縮できます (ただし、データが破損した場合は、バックアップに依存する必要があり、通常は復旧時点がゼロ以外になります)。ホットスタンバイでは、アクティブ/パッシブ設定を使用し、ユーザーは単一のリージョンにのみ誘導され、DR 用リージョンにはトラフィックが送信されません。ほとんどのお客様にとって、第 2 リージョンで完全な環境を立ち上げる場合は、アクティブ/アクティブ設定を使用する方が合理的です。一方、ユーザートラフィックの処理に両方のリージョンを使用したくない場合は、ウォームスタンバイを使用する方がより経済的で、運用上の複雑さも軽減されます。

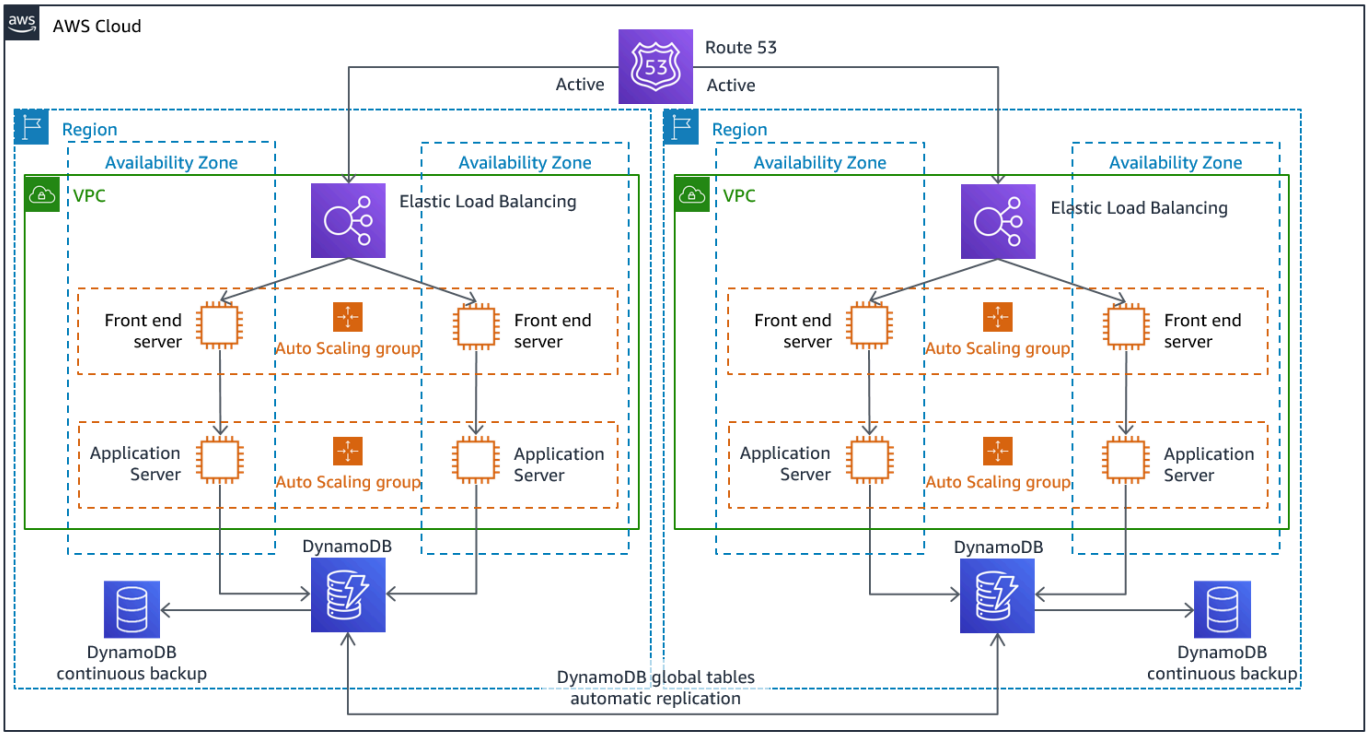

図 12 - マルチサイト アクティブ/アクティブアーキテクチャ (ホットスタンバイの場合は 1 つのアクティブパスを非アクティブに変更)

マルチサイト アクティブ/アクティブの場合、ワークロードは複数のリージョンで実行されるため、このシナリオではフェイルオーバーなどはありません。この場合の災害対策テストでは、リージョンの喪失に対するワークロードの反応に注目します。トラフィックのルーティング先が障害の発生したリージョンから移り、 他方のリージョンですべてのトラフィックを処理できるかを確認します。データ災害に対するテストも必要です。バックアップと復旧は、依然として必要であり、定期的にテストする必要があります。また、データの破損、削除、または難読化を伴うデータ災害の復旧時間は常にゼロより長くなり、復旧点は常に災害の検出前の特定の時点になることにも注意してください。ほぼゼロの復旧時間を維持するために、マルチサイト アクティブ/アクティブ (またはホットスタンバイ) アプローチに伴う追加の複雑さやコストが必要である場合は、セキュリティを維持し、人為的エラーを防止して人為的災害を軽減する追加の工夫が必要になります。

AWS のサービス

ここでも、「バックアップと復元」、「パイロットライト」、「ウォームスタンバイ」で取り上げたすべての AWS のサービスを、データのバックアップ、データのレプリケーション、アクティブ/アクティブのトラフィックルーティング、EC2 インスタンスを含むインフラストラクチャのデプロイとスケーリングに使用します。

前述のアクティブ/パッシブのシナリオ (パイロットライトとウォームスタンバイ) では、Amazon Route 53 と AWS Global Accelerator の両方を使用して、ネットワークトラフィックをアクティブなリージョンにルーティングできます。アクティブ/アクティブ戦略の場合は、これらの両方のサービスを使用して、どのユーザーをどのアクティブなリージョンエンドポイントに誘導するかを決定するポリシーの定義も有効にします。AWS Global Accelerator では、各アプリケーションエンドポイントに送信するトラフィックの割合を制御するトラフィックダイヤルを設定します。Amazon Route 53 は、このパーセンテージアプローチに加えて、地理的近接性やレイテンシーベースのポリシーなど、他の利用可能な複数のポリシーをサポートしています。Global Accelerator は、AWS エッジサーバーの広範なネットワークを自動的に活用し、トラフィックを AWS ネットワークバックボーンにできるだけ早くオンボーディングしてリクエストのレイテンシーを短縮します。

この戦略のデータレプリケーションにより、RPO はほぼゼロになります。Amazon Aurora Global Database などの AWS のサービスでは、専用のインフラストラクチャを使用し、アプリケーションでデータベースを全面的に利用できるようにします。また、1 つのセカンダリリージョンへのレプリケーションに伴うレイテンシーは、一般的に 1 秒未満になります。アクティブ/パッシブ戦略では、書き込みはプライマリリージョンに対してのみ行われます。アクティブ/アクティブとの違いは、アクティブな各リージョンへの書き込みを処理する方法の設計にあります。ユーザーの読み取りは、最寄りのリージョンで処理するように設計するのが一般的です。これはローカルな読み取りと呼ばれます。書き込みには、いくつかのオプションがあります。

- グローバルな書き込み戦略では、すべての書き込みを 1 つのリージョンにルーティングします。そのリージョンに障害が発生した場合は、別のリージョンが昇格されて書き込みを受け入れます。Aurora グローバルデータベースは、グローバルな書き込みに最適です。このデータベースは、リージョン間でのリードレプリカとの同期をサポートし、セカンダリリージョンの 1 つを昇格させて 1 分未満で読み取り/書き込みの責任を引き継げるようにします。
- ローカルな書き込み戦略では、書き込みを最寄りのリージョンにルーティングします (読み取りと同様です)。Amazon DynamoDB グローバルテーブルを使用すると、このような戦略が可能になり、グローバルテーブルのデプロイ先であるすべてのリージョンから読み取りと書き込みができます。Amazon DynamoDB グローバルテーブルは、同時更新間での最終書き込み者優先の調整を行います。
- 書き込みのパーティション化戦略では、書き込みの競合を避けるために、パーティションキー (ユーザー ID など) に基づいて特定のリージョンに書き込みを割り当てます。この場合、双方向に設定された Amazon S3 レプリケーションを使用できます。現在 2 つのリージョン間でのレプリケーションがサポートされています。このアプローチを実装する場合は、バケット A とバケット B の両方でレプリカ変更の同期を必ず有効にし、レプリケートしたオブジェクトに対するオブジェクトアクセスコントロールリスト (ACL)、オブジェクトタグ、オブジェクトロックなどのレプリカメタデータの変更をレプリケートしてください。また、アクティブなリージョンのバケット間で削

除マーカーをレプリケートするかどうかも設定できます。レプリケーションに加えて、ポイントインタイムバックアップも戦略に含め、データの破損や破壊イベントから保護する必要があります。

AWS CloudFormation は、複数の AWS リージョンの AWS アカウント間でインフラストラクチャを一貫してデプロイするための強力なツールです。AWS CloudFormation StackSets は、この機能を拡張し、1 回の操作で複数のアカウントやリージョンにわたって CloudFormation スタックを作成、更新、または削除できるようにします。AWS CloudFormation では YAML または JSON を使用して Infrastructure as Code を定義しますが、AWS Cloud Development Kit (AWS CDK) では、使い慣れたプログラミング言語を使用して Infrastructure as Code を定義できます。コードは CloudFormation に変換された後で、AWS にリソースをデプロイするために使用されます。

検出

ワークロードが本来もたらすべきビジネス上の成果をもたらしていないことを、できるだけ早く知ることが重要です。これにより、障害をすばやく宣言し、インシデントから復旧できます。厳しい復旧目標の場合、この応答時間と適切な情報の組み合わせが、復旧目標の達成に不可欠です。目標復旧時点が 1 時間の場合は、インシデントの検出、適切な担当者への通知、エスカレーションプロセスの実施、DR 計画を実行しない場合の復旧予想時間に関する情報 (ある場合) の評価、災害の宣言、および 1 時間以内での復旧を行う必要があります。

> **Note**
>
> RTO を達成できない危険性があっても、ステークホルダーが DR を実行しないことを決定した場合は、DR 計画と目標を再評価します。DR 計画を実行しないという決定は、計画が不十分であるか、実行に自信が持てないことが原因である可能性があります。

ビジネス価値をもたらす現実的で達成可能な目標を提供するには、インシデントの検出、通知、エスカレーション、検出、宣言を、計画と目標に組み込むことが重要です。

AWS は、Service Health Dashboard でサービスの可用性に関する最新情報を公開しています。いつでも確認して最新のステータス情報を入手したり、RSSフィードを購読して個々のサービスの中断の通知を受けたりすることがきます。AWS のいずれかのサービスでリアルタイムの運用上の問題が発生し、それが Service Health Dashboard に表示されない場合は、サポートリクエストを作成できます。

AWS Health Dashboard には、アカウントに影響する可能性がある AWS Health イベントの情報が表示されます。情報は 2 つの方法で表示されます。ダッシュボードには、最近のイベントおよび予定されているイベントがカテゴリ別に分類されて表示されます。詳細なイベントログには、過去 90 日間のすべてのイベントが表示されます。

最も厳しい RTO 要件に対しては、ヘルスチェックに基づく自動フェイルオーバーを実装できます。重要業績評価指標に基づいて、ユーザーエクスペリエンスを表すヘルスチェックを設計します。詳細なヘルスチェックでは、ワークロードの主要な機能を実行して、簡便なハートビートチェックよりも深く掘り下げて調査します。詳細なヘルスチェックは、複数のシグナルに基づいて使用します。このアプローチでは、誤ったアラームをトリガーしないように注意してください。必要のないときにフェイルオーバーすると、それ自体が可用性のリスクを招くおそれがあります。

災害対策のテスト

災害対策の実装をテストして実装を検証し、ワークロードの DR 用リージョンへのフェイルオーバーを定期的にテストして RTO と RPO が満たされていることを確認します。

回避すべきなのは、めったに実行されない復旧経路を作ることです。たとえば、読み取り専用のクエリに使用されるセカンダリデータストアがあるとします。データストアの書き込み時にプライマリデータストアで障害が発生した場合、セカンダリデータストアにフェイルオーバーします。もしこのフェイルオーバーを頻繁にテストしない場合、セカンダリデータストアの機能に関する前提が正しくない可能性があります。前回のテスト時には十分であったセカンダリの容量が、今回のシナリオでは負荷に耐えられなくなったり、セカンダリリージョンのサービスクォータが十分でなかったりする場合があります。

エラー復旧が有効に機能するのは、頻繁にテストしている復旧経路に限られることが、これまでの経験から明らかです。この理由のため、復旧経路の数を少なくすることが最善です。

復旧パターンを確立して定期的にテストすることができます。復旧経路が複雑または重大な場合は、さらに本番環境で該当する障害を定期的に実行し、復旧経路が正常に機能することを検証する必要があります。

DR 用リージョンで設定ドリフトを管理します。インフラストラクチャ、データ、設定が DR 用リージョンで必要とされる状態であることを確認します。例えば、AMI とサービスクォータが最新であることを確認します。

AWS Config を利用して、AWS リソースの設定を継続的にモニタリングおよび記録できます。AWS Config は、ドリフトを検出して、AWSSystems Manager Automation をトリガーしてドリフトを修正し、アラームを発生させることができます。AWS CloudFormation では、デプロイしたスタックのドリフトも検出できます。

まとめ

クラウド内でのアプリケーションの可用性については、お客様の責任となります。災害とは何かを定義し、この定義とそれがビジネスの成果に与える影響を反映した災害対策計画を立てることが重要です。影響分析とリスク評価に基づいて目標復旧時間 (RTO) と目標復旧時点 (RPO) を作成し、災害を軽減するための適切なアーキテクチャを選択します。災害を適時に検出できることを確認します。どのようなときに目標が危険にさらされるかを知ることは極めて重要です。計画を確実に策定し、テストして検証します。検証していない災害対策計画は、自信を持てないために実装されないか、災害対策目標の達成に失敗するおそれがあります。

寄稿者

本書の寄稿者は以下のとおりです

- Alex Livingstone、AWS Enterprise Support、プラクティスリードクラウドオペレーション
- Seth Eliot、アマゾン ウェブ サービス、プリンシパルリライアビリティソリューションアーキテクト

その他の資料

詳細については、以下を参照してください。

- 信頼性の柱 - AWS Well-Architected Framework
- 災害対策計画のチェックリスト
- ヘルスチェックの実装
- AWS ソリューションの実装: Multi-Region Application Architecture
- AWS re:Invent 2018: Architecture Patterns for Multi-Region Active-Active Applications (ARC209-R2)

ドキュメント履歴

変更	説明	改訂日
初版発行	初版発行。	2021 年 2 月 12 日

このホワイトペーパーの更新に関する通知を受け取るには、RSS フィードをサブスクライブしてください。

注意

お客様は、この文書に記載されている情報を独自に評価する責任を負うものとします。本書は、(a) 情報提供のみを目的とし、(b) AWS の現行製品と慣行について説明しており、これらは予告なしに変更されることがあり、(c) AWS およびその関連会社、サプライヤーまたはライセンサーからの契約上の義務や保証をもたらすものではありません。AWS の製品やサービスは、明示または暗示を問わず、一切の保証、表明、条件なしに「現状のまま」提供されます。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で締結されるいかなる契約の一部でもなく、その内容を修正するものでもありません。